
Homomorphic Factorization of BRDF-based Lighting Computation

ACM SIGGRAPH 2002

San Antonio, Texas, USA

Thursday, 25 July 2002

Lutz Latta and Andreas Kolb

Faculty of Media-Information Science

University of Applied Sciences Wedel, Germany

1. Introduction and Prior Work

2. Lighting Function Factorization

2.1. Approximation Using Homomorphic Factorization

2.2. Sampling the Lighting Function

2.3. Rendering & Implementation Aspects

3. Result

4. Conclusion

1 Introduction

Key Features

Material: Arbitrary isotropic BRDF

Lighting: Arbitrary lighting environment

Pre-Processing: Approximation of lighting function with two textures

Rendering: Realtime & multitexturing

Video

What is a BRDF?

BRDF = Bidirectional Reflectance Distribution Function

Purpose: Relation of outgoing to incoming light per unit area at surface point

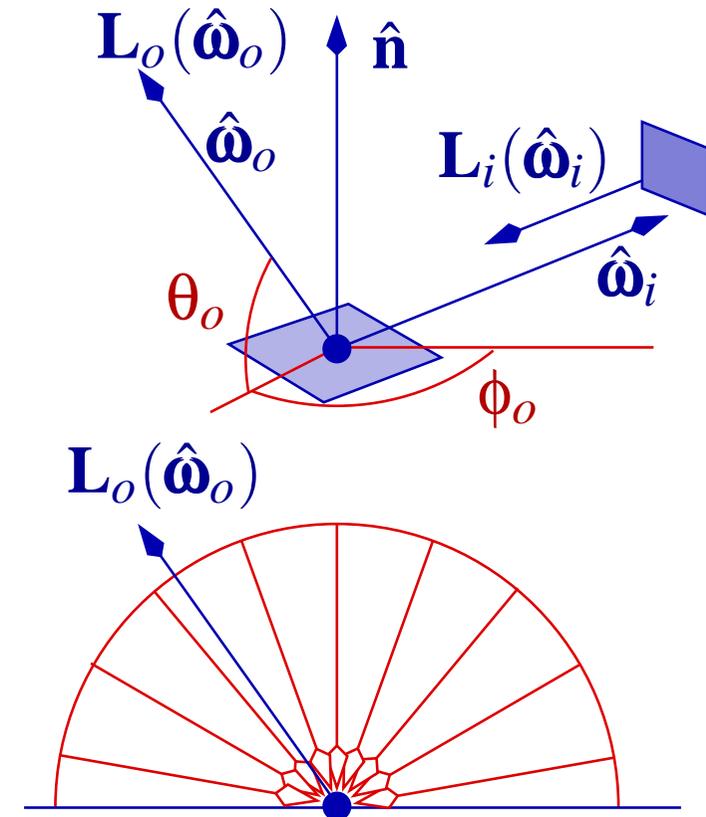
$$BRDF(\hat{\omega}_i, \hat{\omega}_o, \lambda) = \frac{d\mathbf{L}_o(\hat{\omega}_o, \lambda)}{d\mathbf{L}_i(\hat{\omega}_i, \lambda) \cos(\theta_i) d\hat{\omega}_i}$$

- Lambert's Law: $\mathbf{L}_i(\hat{\omega}_i) \cos(\theta_i)$ light per unit area
- λ is represented by RGB color primitives

BRDF-based Lighting

Collect incoming light over hemisphere Ω :

$$\mathbf{L}_o(\hat{\omega}_o) = \int_{\Omega} BRDF(\hat{\omega}_i, \hat{\omega}_o) \mathbf{L}_i(\hat{\omega}_i) \cos(\theta_i) d\hat{\omega}_i$$

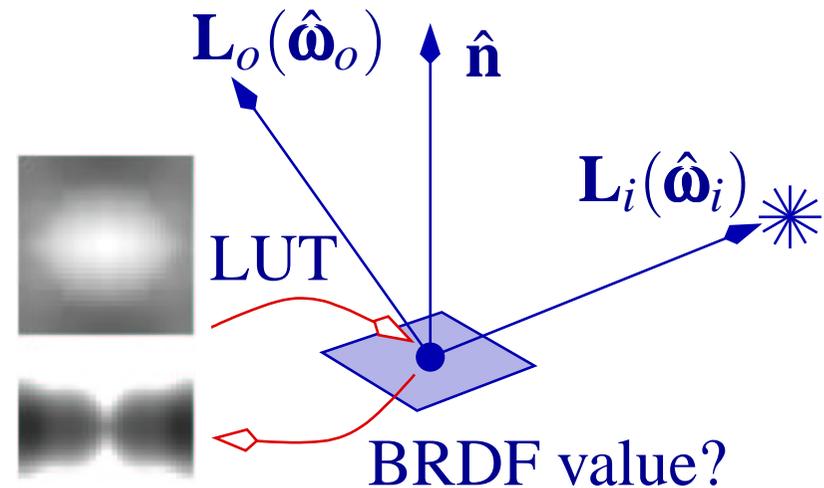


1.1 BRDF-based Lighting in Realtime

BRDF Approximation: e.g.

Kautz&McCool'99, McCool et al.'01

1. Texture maps as BRDF lookup table
2. Small number of light sources

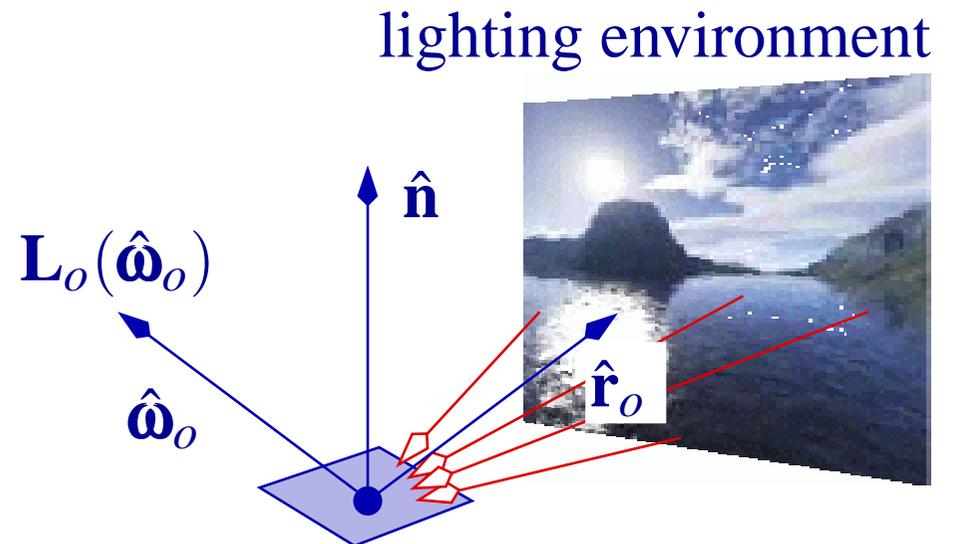


Prefiltered Environment Maps: e.g.

Heidrich&Seidel'99, Kautz&McCool'00

1. Complex lighting environment supplied by environment map
2. Special classes of BRDF simulated using filtering process

Our Approach: Apply BRDF approximation to complete lighting equation



1.2 BRDF Approximation

General Approach: Find space of approximation functions $\{BRDF^*\}$:

1. Efficient determination of best approximation $BRDF^*$ to $BRDF$
2. $BRDF^*$ can be computed in realtime using texture maps t_i

Characteristics: 1. $BRDF^*$ is discrete

2. $BRDF^*$ is computed by solving a linear system of equations

Kautz & McCool: *Singular Value Decomp. (SVD)* yields best approximation

$$BRDF^*(\hat{\omega}_i, \hat{\omega}_o) = \sum_{j=1}^J t_{j,1}(\pi_1(\hat{\omega}_i, \hat{\omega}_o)) t_{j,2}(\pi_2(\hat{\omega}_i, \hat{\omega}_o))$$

π_i : map to texture coord.

McCool et al.: Logarithmic homomorphic transformation

$$BRDF^*(\hat{\omega}_i, \hat{\omega}_o) = t_1(\hat{\omega}_o) t_2(\hat{\mathbf{h}}) t_1(\hat{\omega}_i), \quad \hat{\mathbf{h}} \text{ halfway vector}$$

2 Lighting Function Factorization

General Idea

Approximating the complete lighting function

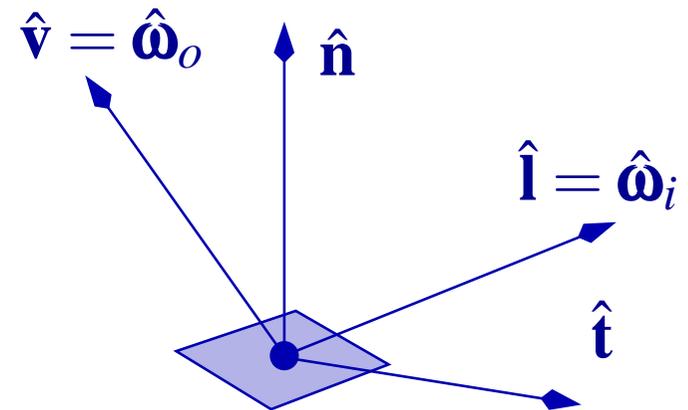
$$\mathbf{L}_o(\hat{\mathbf{v}}, \hat{\mathbf{n}}, \hat{\mathbf{t}}) = \int_{\Omega} BRDF(\hat{\omega}(\hat{\mathbf{l}}, \hat{\mathbf{n}}, \hat{\mathbf{t}}), \hat{\omega}(\hat{\mathbf{v}}, \hat{\mathbf{n}}, \hat{\mathbf{t}})) \mathbf{L}_i(\hat{\mathbf{l}}) (\hat{\mathbf{n}} \cdot \hat{\mathbf{l}}) d\hat{\mathbf{l}}$$

using McCool et al.'s Homomorphic Factorization

$\hat{\omega}(\hat{\mathbf{a}}, \hat{\mathbf{n}}, \hat{\mathbf{t}})$ maps world space vector $\hat{\mathbf{a}}$ to surface coordinates $\{\hat{\mathbf{n}}, \hat{\mathbf{t}}, \hat{\mathbf{n}} \times \hat{\mathbf{t}}\}$

Main differences to approximation of BRDF:

1. Vectors $\hat{\mathbf{v}}, \hat{\mathbf{n}}, \hat{\mathbf{t}}$ represented in world space
2. Vectors $\hat{\mathbf{v}}, \hat{\mathbf{n}}, \hat{\mathbf{t}}$ not restricted to hemisphere
3. Invalid vector combinations e.g. $\angle(\hat{\mathbf{n}}, \hat{\mathbf{v}}) > \frac{\pi}{2}$



Restriction: Isotropic reflection, i.e. tangent vector $\hat{\mathbf{t}}$ not relevant

2.1 Approx. Using Homomorphic Factorization (HF)

Challenge: Need a parameterization in world coordinates!

Simple setup for HF-based approximation of the lighting function (LF):

$$\mathbf{L}_o(\hat{\mathbf{v}}, \hat{\mathbf{n}}) \approx \mathbf{L}_o^*(\hat{\mathbf{v}}, \hat{\mathbf{n}}) = t_1(\hat{\mathbf{n}})t_2(\hat{\mathbf{r}}_{\hat{\mathbf{v}}})$$

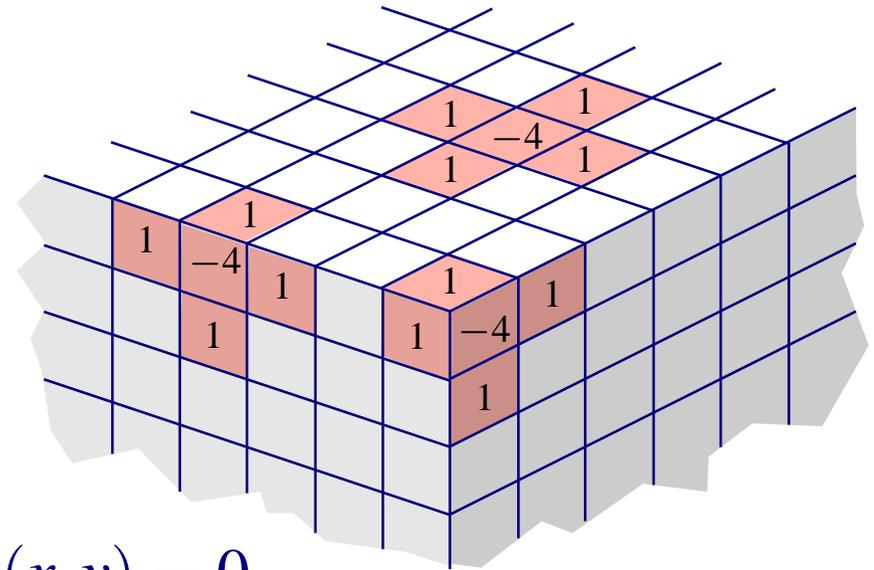
Parameterization: Surface normal $\hat{\mathbf{n}}$ and reflected view vector $\hat{\mathbf{r}}_{\hat{\mathbf{v}}}$

Take Logarithm: With $\bar{a} = \log(a) : \bar{\mathbf{L}}_o^*(\hat{\mathbf{v}}, \hat{\mathbf{n}}) = \bar{t}_1(\hat{\mathbf{n}}) + \bar{t}_2(\hat{\mathbf{r}}_{\hat{\mathbf{v}}})$

Sampling LF yields linear constraints in unknown texel values $t_i(x, y)$

Smoothness Constraints: Laplacian handles unconstrained texels:

$$\begin{aligned} \nabla[t](x, y) &= t(x-1, y) + t(x+1, y) + \\ & t(x, y-1) + t(x, y+1) - 4t(x, y) = 0 \end{aligned}$$



2.2 Sampling the Lighting Function (LF)

LF-samples form the right-hand side of the equation system

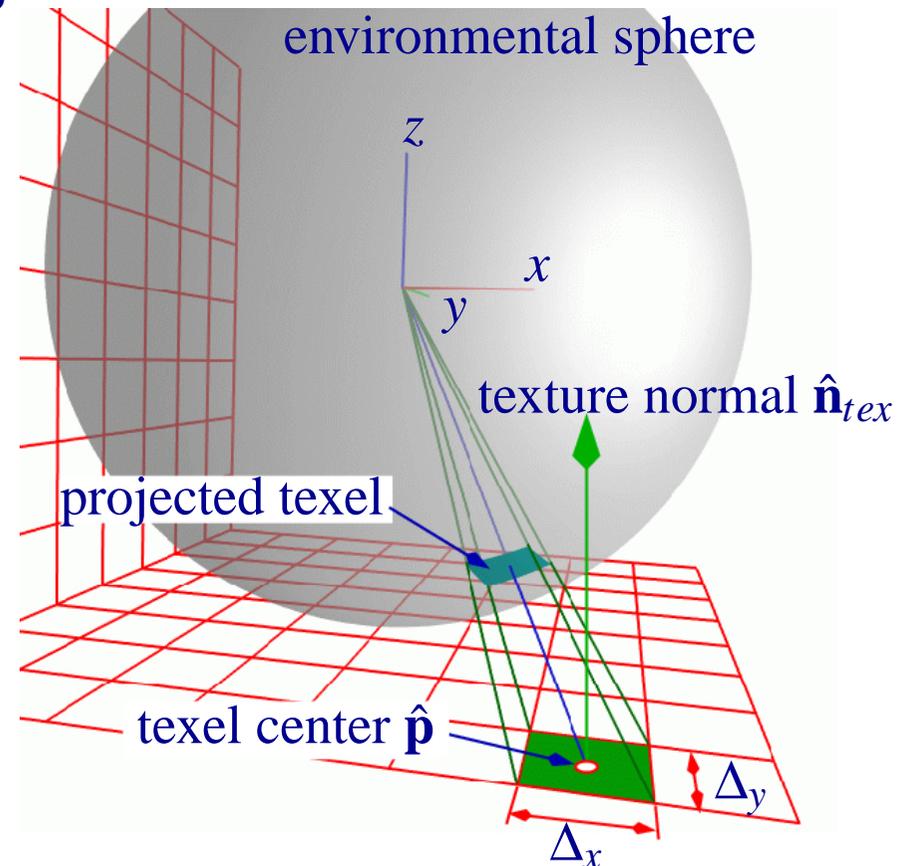
Lighting environment supplied by cube map

$$\mathbf{L}_o^*(\hat{\mathbf{v}}, \hat{\mathbf{n}}) \approx \mathbf{L}_o(\hat{\mathbf{v}}, \hat{\mathbf{n}}) = \frac{1}{\sum_{\text{texel } \hat{\mathbf{p}}} g(\hat{\mathbf{p}})} \sum_{\text{texel } \hat{\mathbf{p}}} g(\hat{\mathbf{p}}) BRDF(\hat{\mathbf{p}}, \hat{\mathbf{v}}) \mathbf{L}_i(\hat{\mathbf{p}}) (\hat{\mathbf{n}} \cdot \hat{\mathbf{p}})$$

Weight $g(\hat{\mathbf{p}})$ for texel $\hat{\mathbf{p}}$:

- Projected texel area on sphere
- For texture at $z = -1$:

$$g(\hat{\mathbf{p}}) \approx \cos(\hat{\mathbf{n}}_{tex}, -\hat{\mathbf{p}}) \frac{\text{texel area}}{\|\hat{\mathbf{p}}\|^2} = \frac{\Delta_x \Delta_y}{\|\hat{\mathbf{p}}\|^3}$$



Alternative Parameterizations

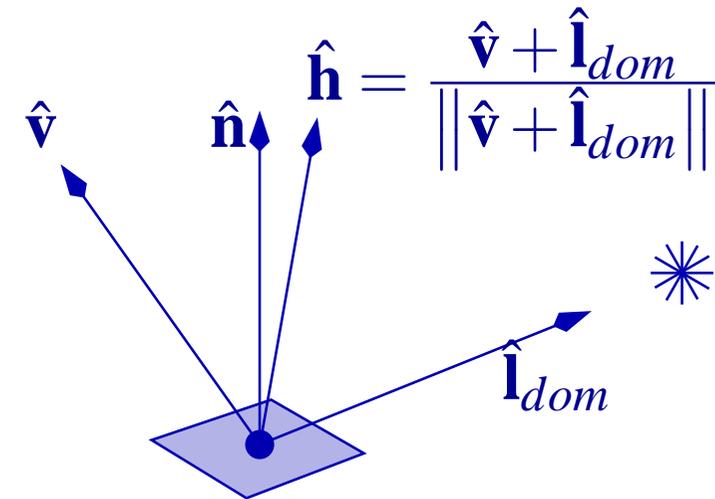
Normal-Reflection-View: $L_o^*(\hat{\mathbf{v}}, \hat{\mathbf{n}}) = t_1(\hat{\mathbf{n}})t_2(\hat{\mathbf{r}}_{\hat{\mathbf{v}}})t_3(\hat{\mathbf{v}})$

Normal-Reflection-Halfway-Difference:

Light comes predominantly from direction $\hat{\mathbf{I}}_{dom}$ (e.g. from the sun).

$$L_o^*(\hat{\mathbf{v}}, \hat{\mathbf{n}}) = t_1(\hat{\mathbf{n}})t_2(\hat{\mathbf{r}}_{\hat{\mathbf{v}}})t_3(\hat{\mathbf{h}})t_4(\hat{\mathbf{d}})$$

$\hat{\mathbf{d}}$ is $\hat{\mathbf{I}}_{dom}$ in surface coordinates



Solving the Linear System

- Large and sparse matrix in unknown texel values
- *Iterative Quasi-Minimal Residual* algorithm (Freund & Nachtigal 1992)

Multitexturing hardware allows one-pass rendering with several textures

Texture Coordinate Computation:

- Normal mapping (t_1) and reflection mapping (t_2) in world space
- Programmable hardware, e.g. vertex shaders

Problem: Computed texture values loose precision when stored in 8bit/channel textures

Pixel pipeline computation:

$$\boxed{\text{constant vertex color}} \cdot t_1(\hat{\mathbf{n}}) \cdot t_2(\hat{\mathbf{r}}_{\hat{\mathbf{v}}}) \cdot \boxed{1, 2 \text{ or } 4}$$

1. Normalize textures and correct with per-vertex color
2. If constant color > 1: post-scaling with factor 2 or 4
3. Manual adjustment of intensity range significant for \mathbf{L}^*

2.3 Rendering & Implementation Aspects

LF-Sampling: • *Back-projection* to regularly constrain texels

- Lighting environment texture resolution: $20^2 - 64^2$ per face

Brute Force LF-Sampling: Sample all texels, e.g. 64^2 per cube face

$$\Rightarrow \frac{1}{2}(64 \times 64 \times 2)^2 \approx 300M \text{ samples}$$

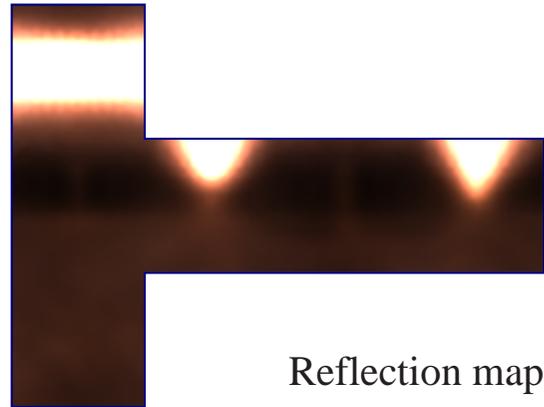
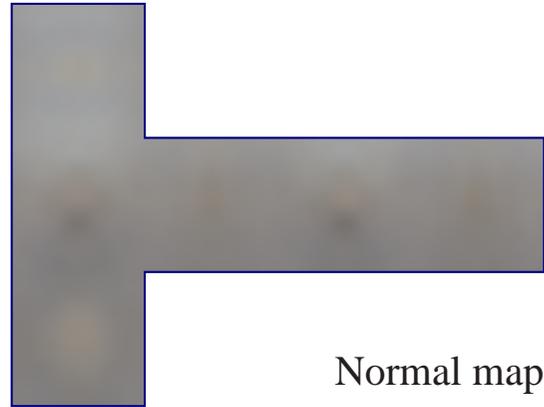
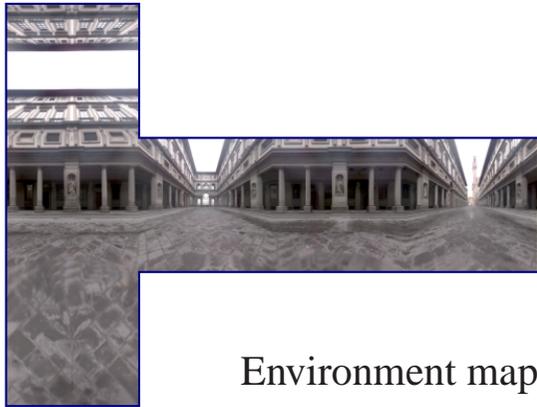
Multi-Resolution Approximation:

1. Start with low texture resolution, usually 16^2 with $12^2 - 16^2$ LF samples
2. Iteration:
 - 2.1. Solve linear system for current resolution
 - 2.2. Refine textures and form new starting solution from old resolution

Approximation Error: Heavily depends upon the concrete data

Runtime: LF-Sampling $15' - 60'$; QMR-solver $10' - 30'$ (AMD Athlon 1.4 GHz)

3 Results



Copper BRDF with Uffizi environment

[Video](#)

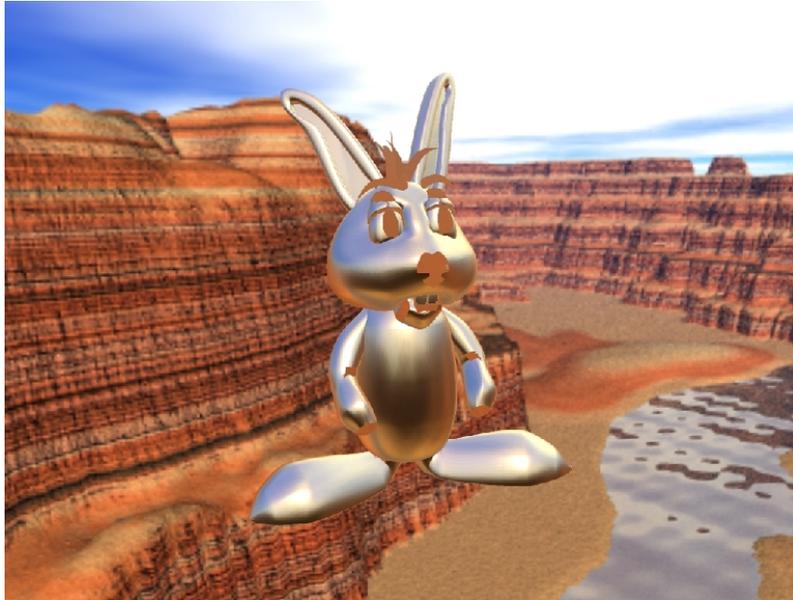
4 Conclusions & Future Work

Conclusions:

- Unified approach for simulating complex lighting and material
- Visually plausible results with acceptable approximation errors
- Relatively high pre-computational costs (LF-sampling and factorization)
- Restricted to isotropic material and static lighting environments

Future Work:

- Introduce anisotropic material (how to parameterize?)
- Reduction of LF-sampling effort using Monte Carlo sampling



Thanks for your attention!

Acknowledgements for BRDF- and environment data:

- Cornell BRDF database
- Jeff Heath
- Paul Debevec